## AMENDMENT TO THE CLAIMS

Please **AMEND** claims 6, 9, 11, 15, 19, and 21 as follows.

A copy of all pending claims and a status of the claims are provided below.

1. (Original) A method for authentication in a network, the method comprising:

creating a credential string which is derived from a session ID;

sending a UserID associated with the session ID and the credential string to a software application;

receiving a confirmation request which includes the credential string; and

sending a response in reply to the confirmation request to validate the credential string to authenticate the UserID.

2. (Original) The method of claim 1, further comprising the step of maintaining a password at a portal and not sending the password to authenticate the UserID.

3. (Original) The method of claim 2, wherein the credential string is an encrypted hash of the session ID.

4. (Original) The method of claim 1, further comprising the steps of:

performing a lightweight directory access protocol (LDAP) lookup using the UserID; and

if the LDAP lookup confirms the UserID and the response validates the credential string, returning a successful authentication reply to the software application for establishing a session associated with the session ID, otherwise sending an unsuccessful authentication reply to the software application.

5. (Original) The method of claim 1, wherein the sending of a UserID and the credential string avoids at least one of sending a user's password outside of a portal server and storing the password in persistent memory.

6. (Currently Amended) The method of claim 1, further comprising the steps of:

sending the UserID associated with the session ID and the credential string to a software application proxy;

checking whether the session ID and the credential string ~~has~~ have been previously received within a predetermined time period; and

if affirmative, initiating a security breach procedure.

7. (Original) The method of claim 6, wherein the security breach procedure causes the termination of any session associated with the UserID.

8. (Original) The method of claim 1, wherein the receiving step and sending a response step is performed by an authentication proxy.

9. (Currently Amended) A method for authenticating a user request for a software application, the method comprising:

receiving a UserID and a credential string at an authentication proxy server, the credential string is derived from a session ID;

sending a confirmation request from the authentication proxy to a portal, the confirmation request includes the credential string;

receiving a response at the authentication proxy for the confirmation request; and validating the UserID using a light weight directory access protocol (LDAP) lookup request and the response.

10. (Original) The method of claim 9, further comprising providing a confirmation to the software application if the response is affirmative and the UserID is authenticated by the LDAP lookup.

11. (Currently Amended) The method of claim 9, further comprising creating the credential string from ~~a~~ the session ID at the portal.

12. (Original) The method of claim 11, further comprising encrypting the credential string.

13. (Original) The method of claim 12, further comprising validating the confirmation request by assuring that the credential string has been received only once for confirmation at the portal, otherwise, if presented more than once, performing at least one of initiating a security breach procedure and notifying a software application proxy.

14. (Original) The method of claim 9, further comprising receiving the UserID and a password during a logon to the portal, wherein the UserID is validated in the validating step and the password is maintained at the portal and used to process the confirmation request.

15. (Currently Amended) A system for authenticating a session stored on a computer readable storage medium, comprising computer readable program code, comprising:

an authentication proxy which receives requests to authenticate a UserID and a credential string; and

a credential string validation component which receives requests to validate the credential string,

wherein the credential string validation component checks whether the credential string has been previously received for validation within a predetermined time period.

16. (Original) The system of claim 15, wherein the authentication proxy performs lightweight directory access protocol (LDAP) lookups using the UserID and sends the credential string to the credential string validation component and receives a validation reply.

17. (Original) The system of claim 16, wherein the authentication proxy sends an affirmative authentication reply to a software application when both the LDAP lookup is successful and the validation reply indicates a valid credential string.

18. (Original) The system of claim 17, wherein the authentication proxy receives the UserID and credential string from a software application.

19. (Currently Amended) The system of claim 15, further comprising a software application proxy which receives the UserID and the credential string and detects whether the UserID and the credential string ~~has~~ have been previously received within a predetermined time period.

20. (Original) The system of claim 19, further comprising a portal to create and encrypt the credential string by hashing a session ID, the portal sends the credential string and the UserID to the software application proxy, and does not send a password associated with the UserID.

21. (Currently Amended) The system of claim 15, further comprising:

a portal for accepting a logon by a user and for creating the credential string from an associated session ID;

a lightweight directory access protocol (LDAP) directory for authenticating the UserIDs and which is accessible by the authentication proxy; and a software application proxy for intercepting the UserID and the credential string sent by the portal for monitoring duplicate occurrences of the UserID and the credential string.

22. (Original) A computer program product comprising a computer usable medium having readable program code embodied in the medium, the computer program product including at least one program code to:

create a credential string which is derived from a session ID;

send a UserID associated with the session ID and the credential string to a software application;

receive a confirmation request which includes the credential string; and

send a response in reply to the confirmation request to validate the credential string to authenticate the UserID.